Computer Science

Frank Appiah

Simulation Analysis Using Matlab

Professorial Dissertation

GRIN

YOUR KNOWLEDGE HAS VALUE

- We will publish your bachelor's and master's thesis, essays and papers

- Your own eBook and book - sold worldwide in all relevant shops

- Earn money with each sale

Upload your text at www.GRIN.com and publish for free

GRIN

Bibliographic information published by the German National Library:

The German National Library lists this publication in the National Bibliography; detailed bibliographic data are available on the Internet at http://dnb.dnb.de .

This book is copyright material and must not be copied, reproduced, transferred, distributed, leased, licensed or publicly performed or used in any way except as specifically permitted in writing by the publishers, as allowed under the terms and conditions under which it was purchased or as strictly permitted by applicable copyright law. Any unauthorized distribution or use of this text may be a direct infringement of the author's and publisher's rights and those responsible may be liable in law accordingly.

Imprint:

Copyright © 2019 GRIN Verlag
ISBN: 9783346031174

This book at GRIN:

https://www.grin.com/document/495711

Frank Appiah

Simulation Analysis Using Matlab

GRIN Verlag

GRIN - Your knowledge has value

Since its foundation in 1998, GRIN has specialized in publishing academic texts by students, college teachers and other academics as e-book and printed book. The website www.grin.com is an ideal platform for presenting term papers, final papers, scientific essays, dissertations and specialist books.

Visit us on the internet:

http://www.grin.com/

http://www.facebook.com/grincom

http://www.twitter.com/grin_com

PROFILE ANALYSIS USING MATLAB.

PROF. FRANK APPIAH, AKA

KING'S COLLEGE LONDON

DEPARTMENT OF COMPUTER SCIENCE AND INFORMATICS,

ENGLAND, UK.

ALUMNI / SOFTWARE GROUP.

OTHER BOOKS

TITLE: Cybernetics: Cat-Mice System http://*HYPERLINK "http://lulu.com/"lulu.com*

Ebook, *Paperback and Hardcover (2019).*

TITLE: Simulation and Modeling: Cat-Mice System http://kdp.amazon*HYPERLINK "http://kdp.amazon.com/".com* , http://lulu.com

Ebook, *Paperback and Hardcover (2019).*

TITLE: Cybernetics Intelligence:Natural System http://kdp.amazon*HYPERLINK "http://kdp.amazon.com/".com* , http://lulu.com , http://blurb.com

Ebook, *Paperback and Hardcover (2019).*

TITLE: Cybercat Simulation in Python http://kdp.amazon*HYPERLINK "http://kdp.amazon.com/".com* , http://lulu.com

Ebook, *Paperback and Hardcover (2019).*

FRANK APPIAH, Professor.

Author:

Prof. Frank Appiah hold studentships at the Bsc. (Bachelor in computer engineering, *Kwame Nkrumah University of Science and Technology*) in 2008, Msc.(software engineering, *King's College London*) in 2009/10 and 2 doctorates of philosophy (DPhil / PhD) in computer engineering from *Kwame Nkrumah University of Science and Technology in 2013.*

Preface

This book is about data analysis on profile data captured from the profile run of a cyberintelligence simulation[1] program called cyberintell. Profile run is about running a state-based task or process switch in a procedure with arguments at run-time. The program on execution takes the state-based task with its arguments and automatcally switches to the exact process to execute. The profile module of C/C++[4] is used in the scripting language, Python[2]. This book considers about 13 tasks of the state-based processes in the simulation of the cyberintelligence program. A randomized and structured inputs are used to automate the process of simulation without interruption in modeling the profile statisitics. The un-interruptive string model used in the approach of simulation is based on randomized inputs-made up of integers or strings.

The randomized integers are programmed in the simulation program to select or subselect other program paths to be able to cover all the tasks of each state-based process. This is a field of simulation subject. The profile module has a stimulate procedure in C/C++, that means profling is as stimulating but stimulating is as profile running. Generally, stimulating is a call on the profile module to run a procedure with a specific state-based task in mind. A profile call is made up of a specific procedure name, here a computer definition in Python language, then a string concatenation of state-based task argument and a comma-seperated data input list from the shell or command line.

The main program is where the profile module is called. The main computer defintion is commented because the profile module creates its own main without joining the main definition running thread. A sampling time called protime is initialized on a randomized choice between three different methods namely sampling random range, ranging random value and a fixed value. The code of the main process of the program execution is shown in the Introduction section.

Profile Contents

Preface.....6
INTRODUCTION.....7
PROGRAM RUN 1 Check Transition Function Profile Analysis.....11
PROGRAM RUN 2 Check Transition Point Profile Analysis.....23
PROGRAM RUN 3 Print Transition Profile Analysis.....39
PROGRAM RUN 4 Show Stored View Profile Analysis.....56
PROGRAM RUN 5 Show Cause-Effect Model Profile Analysis...127
PROGRAM RUN 6 Natural Hierarchy Profile Analysis.....135
PROGRAM RUN 7 Show Cause-Effect Case Profile Analysis.....152
PROGRAM RUN 8 Show Book Summary & Chapters Profile Analysis..167
PROGRAM RUN 9 Show Main Menu Profile Analysis.....175
PROGRAM RUN 10 Validate & Verfiy Profile Analysis.....183
PROGRAM RUN 11 Cyber-Simulation Profile Analysis.....205
PROGRAM RUN 12 Transition Semantics Profile Analysis.....258
PROGRAM RUN 13 Logic Program Profile Analysis.....276
REFERENCES.....316
INDEXES.....320

Introduction

Profile Analysis using Matlab[6] is the analysis of the profile data captured from the profile run of a program using Matlab computing[7] and programming facility. It takes into consideration 11 data analysis functions along with figure generated charts from the profile data from each state-based process. The computer profile of the machine is as follows:

Processor : Intel Pentium 4
Frequency: 1GHz
Memory: 1Gb DDRAM
Architecture: 586
Operating System: Fedora 7 Red Hat Linux[5]

The simulation profile of cyberintelligence program is as follows:

Main Themes: 13
Ideas in Each Theme: 1-12
Profile Runs in Each Theme: 1-12
Profile Charts: 86

Below is the code of the main process:

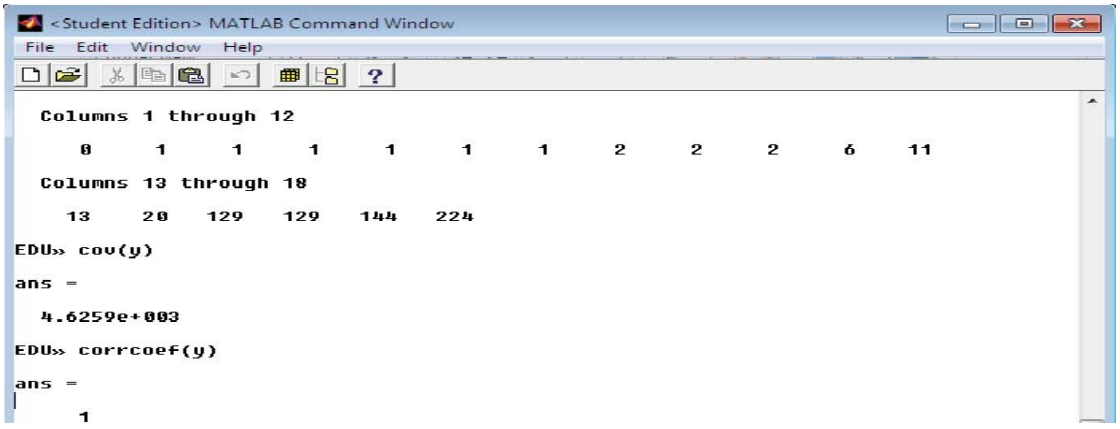
```
if __name__=="__main__":
#def main():
1     Program='readProgram('+sys.argv[1]+', [])';
2     choose=random.choice([1,2,3]);
3     if(choose==1):#sampling time.
4          sample=random.sample(xrange(3), 2);
5          protime=sample[random.randrange(2)];
6     elif(choose==2):
7          protime=random.randrange(2);
8     else:
9          if(int(sys.argv[2])<0):
```

```
10              printLine( "Please provide protime argument value in range(1-
11  10).")
12              protime=int(sys.argv[2]);
13   profile.run(Program);
```

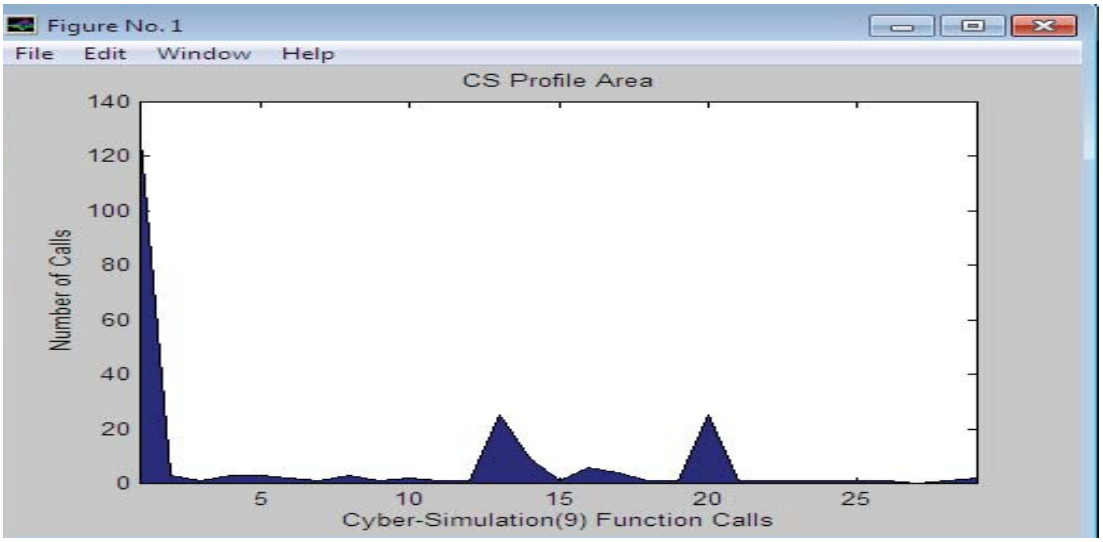
The first line is the program profile to be runned during the execution process. Line 2 is the randomized choice to select a sampling time. Line 3 is a selection of the first sampling time based on the sampling of random ranges. In line 4, it is a generation of the samples ranging from 0 to 3 but only a size of two is needed. Line 5 is a selection of one of the samples from index 0 to index 1. Line 6 is the second selection. Line 7 is to create a randomized value from 0 to 2. In cases, the choice of sampling method is neither 1 or 2 then Line 9 checks for a fixed value from argument. Line 12 does the assignment of the second system argument. The profile running of the simulation is in Line 13.

Now, the simulation program is running. How do we capture the profile data? Well, the profile module already has a profile statistics detail after every execution. Just copy the from the shell or pass in a file name to save (binary though). It is copied into a text file for later analysis. Matlab is the software used in analysing the profile data.

The screenshot of a Windows XP Matlab Software running is shown below:



The figure window of the Matlab plot facility is shown below:



Cisrpro Simulation Test

This simulation has two main programs/procedures:

1. Query Database.
2. Show String Model.

(1) Query Database: It is a program to provide a means to search a cyberintell database.
(2) Show String Model: It is a program that provides a list of store strings that helps in the selection of a particular view of interest.

Installing
---
(1) Check in the folders for the files.
(2) Open a terminal in the folder by pressing F4.
(3) Type bash installdb.sh or sh installdb.sh
(4) Type your mysql password or change username and password,if defaults -u root and -p are used then no change to the installdb file.
(3) Type python setup.py install | python2 setup.py install | python{version_no} setup.py install
(4) Create a cyberintell program: Type "python" on the terminal

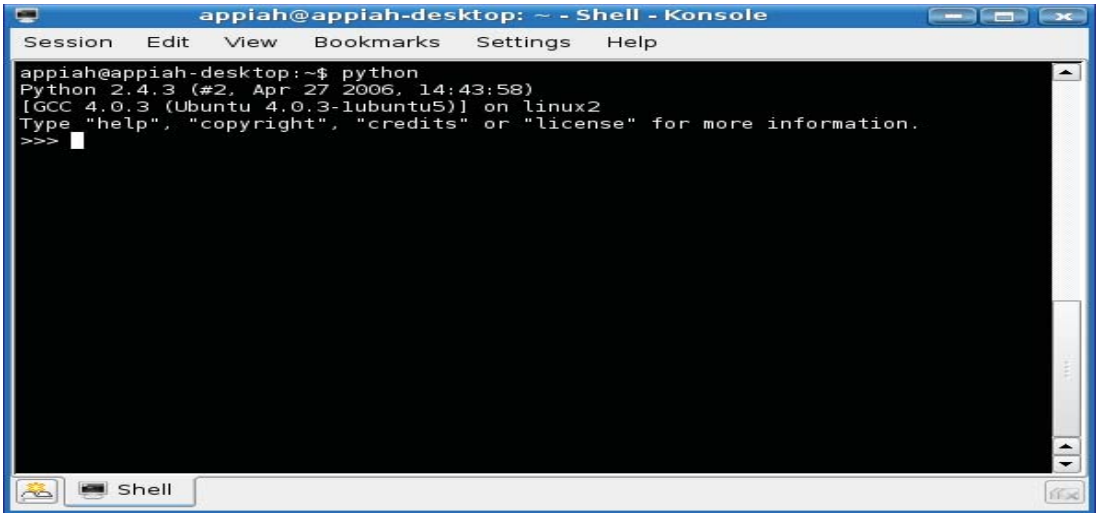
import cisrpro;

(5) (4) or Other Run "sh run.sh" at terminal and follow the prompt discussions on the simulation.
(6) Typical ouput:

```
=======================================
WELCOME TO CYBERINTELL SIMULATION.
WRITTEN BY Dr. Frank Appiah
YEAR: 2019  MONTH: MAY, 1
Version: 1.0.0
=======================================
```

Using cisrpro.pyc; Python Compiled File

Once, you have python install (Windows .exe or Linux .bin), you can test the program path on shell or command line by typing python. If properly installed, the python shell will show up as below (Ubuntu Screenshot):


appiah@appiah-desktop: ~ - Shell - Konsole
Session Edit View Bookmarks Settings Help
appiah@appiah-desktop:~$ python
Python 2.4.3 (#2, Apr 27 2006, 14:43:58)
[GCC 4.0.3 (Ubuntu 4.0.3-1ubuntu5)] on linux2
Type "help", "copyright", "credits" or "license" for more information.
>>>
Shell

Simulate by typing on a shell :

python cisrpro.pyc sim_no delay

Possible sim_no: 1 to 20 Possible delay: 1 to 3

PROGRAM RUN 1 Check Transition Function Profile Analysis.

1.0 Introduction

Check Transition Functions. This program helps one to study the transition functions of a cause-effect(ce) model. Once you know the 9 transition states in ce model of a cyberintel, you can verify a state function of a transition.

1.1 Profile Run

The profile run of check transition function program is shown below:

```
[cyberintell]# python cisrpro.py 1 2

Loaded: Cats File
Loaded: Cyberdata File
Loaded: Simulation File

Start Time=315988818.514
End Time=315988818.514
Execution Time:1.96695327759e-06
CPU time:0.05
Building cause-effect pair simulation inputs.
Done!Building cause-effect simulation inputs.

___Which transition (In lowercases Ex:
 cat=>potential, cause=>effect)?
```

```
>>


____Accepting:$6(1)
cat=>potential
__Checking transition state  1:  cat=> potential

__Transition Point: transit(cat, potential)
____Accepting:$6(1)
potential=>chase
__Checking transition state  1:  cat=> potential

__Final:No transition match for   potential=>chase
__Checking transition state  2:  potential=> search

__Final:No transition match for   potential=>chase
__Checking transition state  3:  search=> human

__Final:No transition match for   potential=>chase
__Checking transition state  4:  potential=> chase

__Transition Point: transit(potential, chase)
____Accepting:$6(1)
chase=>mice
__Checking transition state  1:  cat=> potential

__Final:No transition match for   chase=>mice
__Checking transition state  2:  potential=> search

__Final:No transition match for   chase=>mice
__Checking transition state  3:  search=> human

__Final:No transition match for   chase=>mice
__Checking transition state  4:  potential=> chase

__Final:No transition match for   chase=>mice
__Checking transition state  5:  chase=> mice
```

```
__Transition Point: transit(chase, mice)
____Accepting:$6(1)
cat=>potential
__Checking transition state  1:  cat=> potential

__Transition Point: transit(cat, potential)
____Accepting:$6(1)
potential=>search
__Checking transition state  1:  cat=> potential

__Final:No transition match for   potential=>search
__Checking transition state  2:  potential=> search

__Transition Point: transit(potential, search)
____Accepting:$6(1)
search=>human
__Checking transition state  1:  cat=> potential

__Final:No transition match for   search=>human
__Checking transition state  2:  potential=> search

__Final:No transition match for   search=>human
__Checking transition state  3:  search=> human

__Transition Point: transit(search, human)
____Accepting:$6(1)
cat=>potential
__Checking transition state  1:  cat=> potential

__Transition Point: transit(cat, potential)
____Accepting:$6(1)
potential=>symbol
__Checking transition state  1:  cat=> potential

__Final:No transition match for   potential=>symbol
__Checking transition state  2:  potential=> search
```

```
__Final:No transition match for   potential=>symbol
__Checking transition state  3:  search=> human

__Final:No transition match for   potential=>symbol
__Checking transition state  4:  potential=> chase

__Final:No transition match for   potential=>symbol
__Checking transition state  5:  chase=> mice

__Final:No transition match for   potential=>symbol
__Checking transition state  6:  potential=> symbol

__Transition Point: transit(potential, symbol)
____Accepting:$6(1)
symbol=>operation
__Checking transition state  1:  cat=> potential

__Final:No transition match for   symbol=>operation
__Checking transition state  2:  potential=> search

__Final:No transition match for   symbol=>operation
__Checking transition state  3:  search=> human

__Final:No transition match for   symbol=>operation
__Checking transition state  4:  potential=> chase

__Final:No transition match for   symbol=>operation
__Checking transition state  5:  chase=> mice

__Final:No transition match for   symbol=>operation
__Checking transition state  6:  potential=> symbol

__Final:No transition match for   symbol=>operation
__Checking transition state  7:  symbol=> operation

__Transition Point: transit(symbol, operation)
```

```
____Accepting:$6(1)
operation=>intelligence
__Checking transition state  1:  cat=> potential

__Final:No transition match for   operation=>intelligence
__Checking transition state  2:  potential=> search

__Final:No transition match for   operation=>intelligence
__Checking transition state  3:  search=> human

__Final:No transition match for   operation=>intelligence
__Checking transition state  4:  potential=> chase

__Final:No transition match for   operation=>intelligence
__Checking transition state  5:  chase=> mice

__Final:No transition match for   operation=>intelligence
__Checking transition state  6:  potential=> symbol

__Final:No transition match for   operation=>intelligence
__Checking transition state  7:  symbol=> operation

__Final:No transition match for   operation=>intelligence
__Checking transition state  8:  operation=> intelligence

__Transition Point: transit(operation, intelligence)
____Accepting:$6(1)
intelligence=>physicale
__Checking transition state  1:  cat=> potential

__Final:No transition match for   intelligence=>physicale
__Checking transition state  2:  potential=> search

__Final:No transition match for   intelligence=>physicale
__Checking transition state  3:  search=> human

__Final:No transition match for   intelligence=>physicale
```

```
__Checking transition state  4:  potential=> chase

__Final:No transition match for   intelligence=>physicale
__Checking transition state  5:  chase=> mice

__Final:No transition match for   intelligence=>physicale
__Checking transition state  6:  potential=> symbol

__Final:No transition match for   intelligence=>physicale
__Checking transition state  7:  symbol=> operation

__Final:No transition match for   intelligence=>physicale
__Checking transition state  8:  operation=> intelligence

__Final:No transition match for   intelligence=>physicale
__Checking transition state  9:  intelligence=> physicale
__Transition Point: transit(intelligence, physicale)
```

1.2 Profile Statistics and Analysis

In this section, I will look at the function call statistics, profile charts and profile analysis along with some matlab code used in generating the data analysis objects.

1.2.1 Function Call Statistics

```
Start Time=315988818.514
End Time=315988818.514
```

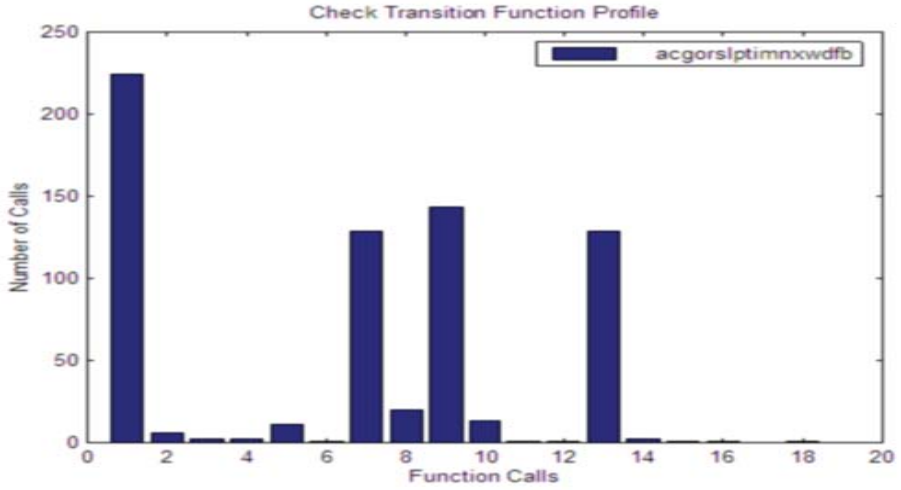
```
Execution Time:1.96695327759e-06
CPU time:0.05
         688 function calls in 0.000 CPU seconds

   Ordered by: standard name

   ncalls  tottime  percall  cumtime  percall filename:lineno(function)
      224    0.000    0.000    0.000    0.000 :0(append)-a
        6    0.000    0.000    0.000    0.000 :0(clock)-c
        2    0.000    0.000    0.000    0.000 :0(getcwd)-g
        2    0.000    0.000    0.000    0.000 :0(open)-o
       11    0.000    0.000    0.000    0.000 :0(range)-r
        1    0.000    0.000    0.000    0.000 :0(setprofile)-s
      129    0.000    0.000    0.000    0.000 :0(sleep)-l
       20    0.000    0.000    0.000    0.000 :0(split)-p
      144    0.000    0.000    0.000    0.000 :0(strip)-t
       13    0.000    0.000    0.000    0.000 :0(time)-i
        1    0.000    0.000    0.000    0.000 <string>:1(<module>)-m
        1    0.000    0.000    0.000    0.00 cisrpro.py:1693(readDataCSV)-d
      129    0.000    0.000    0.000    0.00 cisrpro.py:2078(printLine)-n
        2    0.000    0.000    0.000    0.00 ccisrpro.py:2119(exectime)-x
        1    0.000    0.000    0.000    0.00 cisrpro.py:697(raw_ceinputs)-w
        1    0.000    0.000    0.000    0.000 cisrpro.py:749(readProgram)-d
        0    0.000             0.000          profile:0(profiler)-f
        1    0.000    0.000    0.000    0.0 profile:0(readProgram(1, []))-b
```

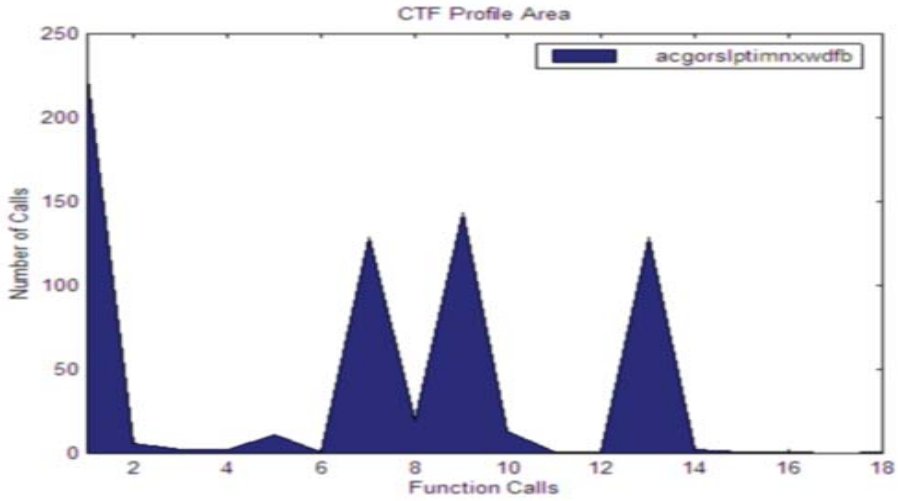
1.2.2 Profile Charts

The check transition function profile chart comprises of a bar chart and an area chart. The label of the x-axis shows the enumerated function calls but can be correspondingly read against the function calls on runs in Section 1.2.1 and the y-axis label is the number of calls for each particular function call on run.

Here, there are two main charts runned on the function calls namely, bar and area charts.The

figures are as shown below:





1.2.3 Profile Analysis

There are about 11 data analysis functions to be performed on the function call statistics. The data analysis performed using Matlab code enables the following functions to calculate these mathematical or statistical objects.

- `Cummulative Sum`

- Mean
- Sum
- Standard Deviation
- Median
- Minimum
- Maximum
- Sort
- Differrence
- Correlation Coefficients and
- Covariance.

I will start with the cummulative sum and three others for this program run.

1.2.3.1 Function Calls Cummulative Sum

```
cumsum(y)

ans =

  Columns 1 through 12

   224   230   232   234   245   246   375   395   539   552
     553   554

  Columns 13 through 18

   683   685   686   687   687   688
```

1.2.3.2 Function Call Sorting

```
sort(y)

ans =

  Columns 1 through 12
```

```
     0     1     1     1     1     1     1     2     2     2          6    11

  Columns 13 through 18

    13    20   129   129   144   224
```

1.2.3.3 Covariance

```
cov(y)

ans = 4.6259e+003
```

1.2.3.4 Correlation Coefficient

```
corrcoef(y)

ans =  1
```

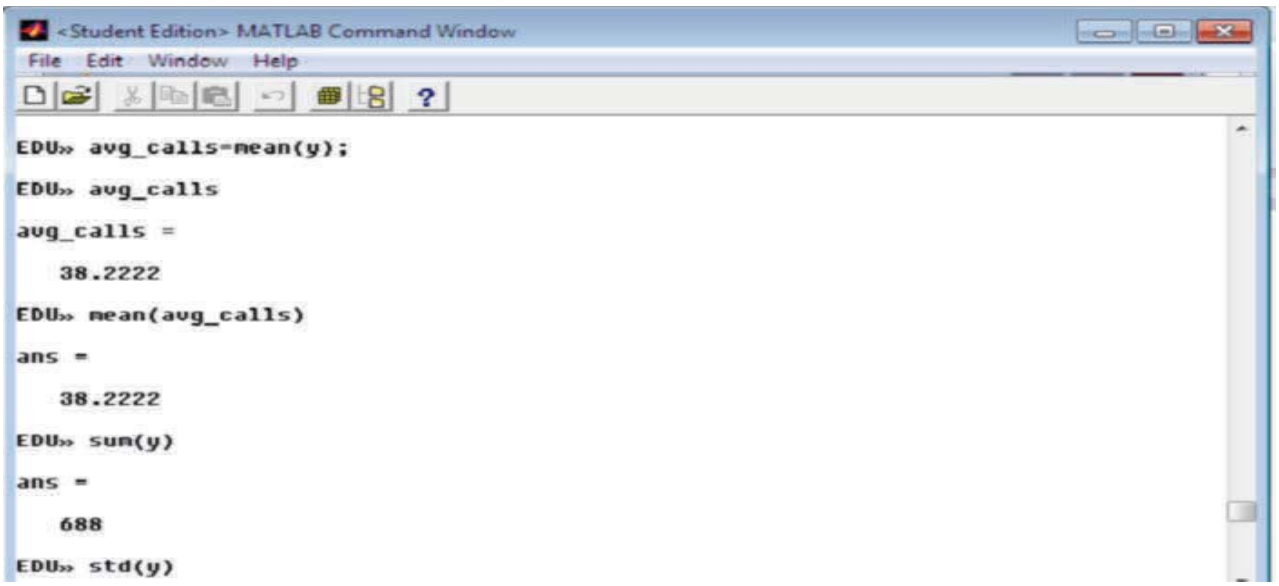
1.3 Matlab Code (M-File)

This is the complete matlab code used in generating and analyzing the profile data on the program run or profile run.

```
%Profile Plot
x=['a','c','g','o','r','s','l','p','t','i','m','d','n','x','w','d','f','b'];
x1=0.1:0.1:1.8;
y=[224,6,2,2,11,1,129,20,144,13,1,1,129,2,1,1,0,1];
plot(x1,y);
xlabel('Function Calls');
ylabel('Number of Calls');
title('Check Transition Function (CTF) Profile');
legend(x);
area(x1,y);
xlabel('Function Calls');
ylabel('Number of Calls');
title('CTF Profile Area');
loglog(x1,y);
```

```
M=peak(224);
subplot(M);
title('Check Transition Function Peak');
disp(['Lengend Meaning']);
disp(['append(a), ']);

%Profile Analysis

overall_avg=mean(avg_calls); %average of calls.
total_calls=sum(y);
std_calls=std(y);  % standard deviation of calls.
med_calls=median(y);
min_calls=min(y);
max_calls=max(y);
sort_calls=sort(y); %ascending sort of calls.
dif_calls=diff(y); %difference of elements iin function calls.
cumsum(y);    %cummulative sum of calls.
corrcoef(y); %correlation coefficients of calls.
cov_calls=cov(y); %covariance of function calls.
```

A screenshot of a Matlab running the code:



1.4 Summary

The objective of this text took an approach on profile analysis of the profile run with short programs written in Matlab's language constructs. The data analysis toolbox in Matlab Student Edition 5 was used in carrying out this profile analysis. In

the sections, I look at an introdcutory text of the check transition function, then to profile run on the program, then thirdly profile analysis and statistics- function statistics, charts and analysis and finally the Matlab code used in the approach.

PROGRAM RUN 2 Check Transition Point Profile Analysis.

2.0 Introduction

Check Transition Points : This program helps one to study the transition points of a cause-effect model. Once you know about 19 or more transition path in ce model of a cyberintel, you can verify a path of model.

2.1 Profile Run

The profile run of check transition points program is shown below:

```
[cyberintell]$ python cisrpro.py 2 1

Loaded: Cats File
Loaded: Cyberdata File


Start Time=315989598.994
End Time=315989598.994
Execution Time:5.00679016113e-06
CPU time:0.09
Building cause-effect pair simulation inputs.
Done!Building cause-effect simulation inputs.

Simulation 2:Check Transition Points  Session@315989599.002


___Which transition path(In lowercases Ex:
 cat=>potential,  cause=>effect)?

>>
Building cepath simulation inputs.
Done!Building cepath simulation inputs.


__Accepting path:cat=>chase
```

```
cat=>chase

___Which Interaction?
 (a)Cat-Mice Interaction
 (b)Cat-Human Interaction
 (c)Cat-Physicale Interaction

>>>
Interaction Simulated is a

___Selection: Cat-Mice Interaction
__Transition Point:  transit(cat, chase)
__End: Transition points checked in cat=>...=>mice

Start Time=315989598.994
End Time=315989598.994
Execution Time:5.00679016113e-06
CPU time:0.1


__Accepting path:cat=>chase
cat=>chase

___Which Interaction?
 (a)Cat-Mice Interaction
 (b)Cat-Human Interaction
 (c)Cat-Physicale Interaction

>>>
Interaction Simulated is a

___Selection: Cat-Mice Interaction
__Transition Point:  transit(cat, chase)
__End: Transition points checked in cat=>...=>mice

Start Time=315989598.994
End Time=315989598.994
Execution Time:5.00679016113e-06
CPU time:0.1


__Accepting path:potential=>mice
potential=>mice

___Which Interaction?
 (a)Cat-Mice Interaction
 (b)Cat-Human Interaction
 (c)Cat-Physicale Interaction

>>>
Interaction Simulated is a

___Selection: Cat-Mice Interaction
__Transition Point:  transit(potential, mice)
__End: Transition points checked in cat=>...=>mice
```

```
Start Time=315989598.994
End Time=315989598.994
Execution Time:5.00679016113e-06
CPU time:0.1


__Accepting path:cat=>mice
cat=>mice

___Which Interaction?
 (a)Cat-Mice Interaction
 (b)Cat-Human Interaction
 (c)Cat-Physicale Interaction

>>>
Interaction Simulated is a

___Selection: Cat-Mice Interaction
__Transition Point:  transit(cat, mice)
__End: Transition points checked in cat=>...=>mice

Start Time=315989598.994
End Time=315989598.994
Execution Time:5.00679016113e-06
CPU time:0.1


__Accepting path:cat=>search
cat=>search

___Which Interaction?
 (a)Cat-Mice Interaction
 (b)Cat-Human Interaction
 (c)Cat-Physicale Interaction

>>>
Interaction Simulated is a

___Selection: Cat-Mice Interaction
__End: Transition points checked in cat=>...=>mice

Start Time=315989598.994
End Time=315989598.994
Execution Time:5.00679016113e-06
CPU time:0.1


__Accepting path:cat=>search
cat=>search

___Which Interaction?
 (a)Cat-Mice Interaction
 (b)Cat-Human Interaction
 (c)Cat-Physicale Interaction

>>>
```

```
Interaction Simulated is b

___Selection: Cat-Human Interaction
__Transition Point: transit(cat, search)
__End: Transition points checked in cat=>...=>human

Start Time=315989598.994
End Time=315989598.994
Execution Time:5.00679016113e-06
CPU time:0.1


__Accepting path:cat=>human
cat=>human

___Which Interaction?
 (a)Cat-Mice Interaction
 (b)Cat-Human Interaction
 (c)Cat-Physicale Interaction

>>>
Interaction Simulated is b

___Selection: Cat-Human Interaction
__Transition Point: transit(cat, human)
__End: Transition points checked in cat=>...=>human

Start Time=315989598.994
End Time=315989598.994
Execution Time:5.00679016113e-06
CPU time:0.11


__Accepting path:potential=>human
potential=>human

___Which Interaction?
 (a)Cat-Mice Interaction
 (b)Cat-Human Interaction
 (c)Cat-Physicale Interaction

>>>
Interaction Simulated is b

___Selection: Cat-Human Interaction
__Transition Point: transit(potential, human)
__End: Transition points checked in cat=>...=>human

Start Time=315989598.994
End Time=315989598.994
Execution Time:5.00679016113e-06
CPU time:0.11


__Accepting path:cat=>symbol
cat=>symbol
```

```
___Which Interaction?
 (a)Cat-Mice Interaction
 (b)Cat-Human Interaction
 (c)Cat-Physicale Interaction

>>>
Interaction Simulated is b

___Selection: Cat-Human Interaction
__End: Transition points checked in cat=>...=>human

Start Time=315989598.994
End Time=315989598.994
Execution Time:5.00679016113e-06
CPU time:0.12


__Accepting path:cat=>operation
cat=>operation

___Which Interaction?
 (a)Cat-Mice Interaction
 (b)Cat-Human Interaction
 (c)Cat-Physicale Interaction

>>>
Interaction Simulated is c

__Selection: Cat-Physicale Interaction
__Transition Points: transit(cat, operation)
__End: Transition points checked in cat=>...=>physicale

Start Time=315989598.994
End Time=315989598.994
Execution Time:5.00679016113e-06
CPU time:0.12


__Accepting path:cat=>operation
cat=>operation

___Which Interaction?
 (a)Cat-Mice Interaction
 (b)Cat-Human Interaction
 (c)Cat-Physicale Interaction

>>>
Interaction Simulated is c

__Selection: Cat-Physicale Interaction
__Transition Points: transit(cat, operation)
__End: Transition points checked in cat=>...=>physicale

Start Time=315989598.994
End Time=315989598.994
```

```
Execution Time:5.00679016113e-06
CPU time:0.13


__Accepting path:cat=>intelligence
cat=>intelligence

___Which Interaction?
 (a)Cat-Mice Interaction
 (b)Cat-Human Interaction
 (c)Cat-Physicale Interaction

>>>
Interaction Simulated is c

__Selection: Cat-Physicale Interaction
__Transition Points: transit(cat, intelligence)
__End: Transition points checked in cat=>...=>physicale

Start Time=315989598.994
End Time=315989598.994
Execution Time:5.00679016113e-06
CPU time:0.13


__Accepting path:cat=>intelligence
cat=>intelligence

___Which Interaction?
 (a)Cat-Mice Interaction
 (b)Cat-Human Interaction
 (c)Cat-Physicale Interaction

>>>
Interaction Simulated is c

__Selection: Cat-Physicale Interaction
__Transition Points: transit(cat, intelligence)
__End: Transition points checked in cat=>...=>physicale

Start Time=315989598.994
End Time=315989598.994
Execution Time:5.00679016113e-06
CPU time:0.13


__Accepting path:cat=>physicale
cat=>physicale

___Which Interaction?
 (a)Cat-Mice Interaction
 (b)Cat-Human Interaction
 (c)Cat-Physicale Interaction

>>>
Interaction Simulated is c
```